

```

1 "use strict";
2
3 // 1. String concatiation (문자 연쇄)
4 console.log("my" + "cat");
5 console.log("1" + 2);
6 /* `` -> 중간에 ',','',줄 바꿈등이 알아서 나옴(변수도 마찬가지) */
7 console.log(`string literals :
8   `);
9 1 + 2 = ${1 + 2}`);
10
11 // 2. Numeric operators
12 console.log(1 + 1); // add
13 console.log(1 - 1); // subtract
14 console.log(1 / 1); // divide
15 console.log(1 * 1); // multiply
16 console.log(5 % 2); // remainder
17 console.log(2 ** 3); // exponentiation (제곱)
18
19 // 3. Increment and decrement operators
20 let counter = 2;
21
22 const preIncrement = ++counter;
23 /* counter = counter + 1;
24    preIncrement = counter; */
25 console.log(`preIncrement : ${preIncrement}, counter : ${counter}`);
26
27 const postIncrement = counter++;
28 /* postIncrement = counter;
29    counter = counter + 1; */
30 console.log(`preIncrement : ${postIncrement}, counter : ${counter}`);
31
32 // 4. Assignment operators
33 let x = 3;
34 let y = 6;
35 x += y;
36 x -= y;
37 x *= y;
38 x /= y;
39
40 // 5. Comparison operators
41 console.log(10 < 6); // less than
42 console.log(10 <= 6); // less than or equal
43 console.log(10 > 6); // greater than
44 console.log(10 >= 6); // greater than or equal
45
46 // 6. Logical operators : \ (or), & (and), ! (not)
47 const value1 = false;
48 const value2 = 4 < 2;
49
50 // \ (or) -> 세개 중에 하나라도 true 면 true 리턴
51 // or -> finds the first truthy value -> 첫번째 값이 true 면 연산 종료
52 // 실무에선 check() 같은 무거운 함수등을 앞에 두지 않음!(간단한 변수를 앞으로)
53 console.log(`or : ${value1 || value2 || check()}`);
54
55 function check() {
56   for(let i = 0; i < 10; i++ ) {
57     //wasting time
58     console.log('hi~');
59   }
60   return true;
61 }

```

```

62
63 // && (and) -> 세개 중에 하나라도 false 면 false 리턴
64 // and -> finds the first falsy value -> 첫번째 값이 false 면 연산 종료
65 console.log(`and : ${value1 && value2 && check()}`);
66 function check() {
67     for(let i = 0; i < 10; i++ ) {
68         //wasting time
69         console.log('hi~');
70     }
71     return true;
72 }
73 // 오브젝트의 값이 null 인지 아닌지 체크할 때도 사용(뒤에 연산이 미실행되기 때문에)
74 // often used to compress long if-statement
75 // nullableObject && nullableObject.something
76 /* if (nullableObject != null) {
77     nullableObject.something;
78 } */
79
80 // ! (not) -> 값을 반대로
81 console.log(!value1);
82
83 //7. Equality
84 const stringFive = '5';
85 const numberFive = 5;
86
87 // == -> loose equality, with type conversion -> 타입을 변경해서 검사
88 console.log(stringFive == numberFive); // true
89 console.log(stringFive != numberFive); // false
90
91 // === -> strict equality, no type conversion -> 타입을 변경하지 않고 검사
92 console.log(stringFive === numberFive); // false
93 console.log(stringFive !== numberFive); // true
94
95 // object equality by reference -> 신경써서 작성할 것!!!
96 // object는 메모리에 탑재될 때 reference 형태로 저장됨
97 const ellie1 = {name : "ellie"};
98 const ellie2 = {name : "ellie"};
99 const ellie3 = ellie1;
100 console.log(ellie1 == ellie2); // 다른 레퍼런스를 참조하므로 false
101 console.log(ellie1 === ellie2); // 같은 타입이든 아니든 다른 레퍼런스를 참조하므로 false
102 console.log(ellie1 === ellie3); // 같은 value를 할당하였기 때문에 true
103
104 // equality - puzzler
105 console.log(0 == false);
106 console.log(0 === false); // 0 != boolean type -> false
107 console.log('' == false);
108 console.log('' === false); // '' != boolean type -> false
109 console.log(null == undefined);
110 console.log(null === undefined); // 둘은 다른 타입 -> false
111
112 // 8. Conditional operators : if
113 // if, else if, else
114 const name = "choco";
115 if (name === "choco") {
116     console.log("welcome Choco!!!");
117 } else if (name === "coder") {
118     console.log("You are amazing coder~!");
119 } else {
120     console.log("unkwon user...");
121 }
122

```

```
123 // 9. Ternary operator : ?
124 // condition ? value1 : value2;
125 // 앞에 조건이 true냐? -> 맞으면 x 출력 : 아니면 no 출력
126 // 간단하게 사용할때는 좋지만 nesting 해서 사용하면 코드 가독성이 떨어지니까 조심
127 console.log(name === "choco" ? "yes" : "no");
128
129 // 10. Switch statement
130 // use for multiple if checks
131 // use for enum-like value check
132 // use for multiple type checks is TS
133 const browser = "IE";
134 switch (browser) {
135     case "IE":
136         console.log("go away!");
137         break;
138
139     case "Chrome":
140     case "Firefox": // 출력 값이 같을 경우 이렇게 작성 가능
141         console.log("love you!");
142         break;
143
144     default:
145         console.log("same all~");
146         break;
147 }
148
149 // 11. Loops
150 // while loop, while the condition is truthy,
151 // body code is executed.
152 let i = 3;
153 while (i > 0) { // 조건이 false가 될때까지 무한 반복
154     console.log(`while : ${i}`);
155     i--;
156 }
157
158 // do while loop, body code is executed first,
159 // then check the condition.
160 do { // do 블록을 먼저 실행 후 뒤에 조건이 맞는지 검사
161     console.log(`do while : ${i}`);
162     i--;
163 } while (i > 0);
164
165 // for loop, for(begin; condition; step)
166 // begin -> 맨 처음 실행시에만 실행
167 for (i = 3; i > 0; i--) {
168     console.log(`for : ${i}`);
169 }
170
171 // 조건문안에 지역변수를 설정하여 작성하는 것도 괜찮음
172 for (let i = 3; i > 0; i = i - 2) {
173     // inline variable declaration
174     console.log(`inline variable for : ${i}`);
175 }
176
177 // nested loops -> 컴퓨터에 부담이 가므로 되도록 사용하지는 말 것
178 for (let i = 0; i < 5; i++) {
179     for (let j = 0; j < 5; j++) {
180         console.log(`i : ${i}, j : ${j}`);
181     } /* for in */
182 } /* for out */
183
```

```
184 // break, continue
185 // Q1. iterate from 0 to 10 and print only even numbers (use continue)
186 for (i = 0; i <= 10; i++) {
187     if(i % 2 == 0) {
188         console.log(`i : ${i}`);
189         continue;
190     }
191 }
192
193 // Q1.1 -> 연습문제로 만들었기 때문에 이렇게 작성한 것(원래는 true로 만드는 것이 편함)
194 for (let i = 0; i < 11; i++) {
195     if (i % 2 !== 0) {
196         continue; // console.log 가 출력되지 않도록 처리
197     }
198     console.log(`Q1 : ${i}`);
199 }
200
201 // Q1.2
202 for (let i = 0; i < 11; i++) {
203     if(i % 2 === 0) {
204         console.log(`Q1 i : ${i}`);
205     }
206 }
207
208 // Q2. iterate from 0 to 10 and print numbers until reaching 8 (use break)
209 for (i = 0; i <= 10; i++) {
210     if (i == 8) {
211         break;
212     }
213     console.log(`i : ${i}`);
214 }
215
216 // Q2
217 for (i = 0; i < 11; i++) {
218     if (i > 8) {
219         break;
220     }
221     console.log(`Q2 : ${i}`);
222 }
```